

# FAX TRANSMITTAL COVER SHEET

---

**CONLEY ROSE, P.C.**  
**5601 Granite Parkway, Suite 750**  
**Plano, Texas 75024-6616**  
Fax Number: (972) 731-2289  
Telephone Number: (972) 731-2288

---

---

**PLEASE DELIVER THE FOLLOWING PAGES IMMEDIATELY TO:**

**NAME:** Examiner Ben Wang, Group Art Unit 2192  
U.S. Patent and Trademark Office

**FAX:** 571 270 2240

**FROM:** Brian C. Genco

**DATE:** November 9, 2010

**RE:** U.S. Patent Application No. 10/723,967  
Examiner's Amendment (17 pages)

**REMARKS:**

Total Number of Pages (Including This One): 18  
OUR CLIENT/MATTER NO.: 4000-16100  
YOUR REFERENCE NO.: Patent Application No. 10/723,967

**IF YOU DO NOT RECEIVE ALL THE PAGES,  
PLEASE CALL 972-731-2288 AS SOON AS POSSIBLE**

*This facsimile and the information it contains is intended to be a confidential communication only to the person or entity to whom it is addressed. If you have received this facsimile in error, please notify us by telephone at the above telephone number and return the original to this office by mail.*

**Atty Docket: IDF 2584 (4000-16100)****Patent****IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicant: Joseph G. Laura

Serial No.: 10/723,967

Filed: November 26, 2003

For: APPLICATION MONITOR SYSTEM AND  
METHOD

Group Art Unit: 2192

Examiner: Wang, Ben C.

Confirmation No. 9521

\$  
\$  
\$  
\$  
\$  
\$  
\$  
\$  
\$  
\$  
\$**EXAMINER'S AMENDMENT**

Applicants thank Examiner Wang for their time and consideration in the telephone interview on October 26, 2010 regarding proposed claim amendments to place the application in condition for allowance. In the interview, Examiner Wang indicated that the claims may be allowable based on including limitations directed to a COBOL application, shared memory, and a technical layer that enables the COBOL application to read values from the shared memory into each of the independent claims.

In a follow-up telephone interview on November 9, 2010, Examiner Wang requested clarification for why independent claims 12 and 21 did not recite a shared memory area as it is recited in independent claims 1 and 35. Applicants noted that claims 12 and 21 are directed to the disclosed memory attachment embodiment shown in Fig. 4 and described in the specification starting at paragraph [0044], whereas claims 1 and 35 are directed to the disclosed shared memory area embodiment shown in Fig. 1 and described in the specification starting at paragraph [0018]. As disclosed in paragraph [0018] an application may be programmed or coded to create a shared memory and to write values into variables contained within the shared memory. In contrast, paragraph [0048] discloses that when an executable of an application is run on a computer system,

***Atty Docket: IDF 2584 (4000-16100)******Patent***

the application is loaded into the computer system memory. As noted in paragraph [0049], in the memory attachment embodiment, it is not necessary for an application to mirror the values of internal variables by writing duplicate values in a shared memory area. Therefore the application does not need to be changed to enable the operation of the monitor.

Applicants request the following amendments be entered in an Examiner's Amendment to place the application in condition for allowance. Applicants submit that no new matter is introduced by these amendments. The changes made are shown by underlining the added text and striking through or surrounding with double brackets the deleted text.

**Claims** are reflected in the listing of claims which begins on page 3 of this paper.

*Atty Docket: IDF 2584 (4000-16100)**Patent****Listing of the Claims:***

1. (Currently Amended) A system for non-intrusively monitoring an application, comprising:

~~at least one application stored on a computer readable medium, the at least one application creates a shared memory area and stores application values in the shared memory area;~~

a first COBOL (Common Business Oriented Language) program module stored on a non-transitory computer-readable medium that is programmed to share[[s]] and attach[[es]] through a technical layer to [[the]]a shared memory area created by the application and that is used by the at least one application during real-time operation to store application values in the shared memory area, the first module is further programmed to read[[s]] at least one application value[[s]] from the shared memory area that [[have]]has been stored in the shared memory area by the at least one application during real-time operation, wherein the first COBOL program module is programmed with COBOL that does not provide support for features to share and attach to the shared memory area, and wherein the technical layer comprises a shared memory routine that includes functions for managing shared memory that enable the first COBOL program module to share and attach to the shared memory area;

a second module stored on a non-transitory computer-readable medium in communication with the first COBOL program module that requests the first

*Atty Docket: IDF 2584 (4000-16100)**Patent*

COBOL program module to read the application values, the second module receives the application values from the first module; and  
a third module stored on a non-transitory computer-readable medium in communication with the second module that displays the application values.

2-3. (Canceled)

4. (Currently Amended) The system of Claim 1, wherein the at least one application value[[s]]\_is [[are]] further defined as at least one application variable and a value for the at least one application variable.

5. (Currently Amended) The system of Claim 1, wherein the first COBOL program module is further configured to communicate the application values to the second module in hypertext markup language format.

6. (Original) The system of Claim 1, wherein the third module is further defined as a graphical user interface.

7. (Currently Amended) The system of Claim 6, wherein the graphical user interface is further configured to receive an input identifying the application values to be read and configured to request the application values identified to the first COBOL program module, via the second module, and wherein the first COBOL program module is configured to read the requested application values data from the shared memory area

**Atty Docket: IDF 2584 (4000-16100)**

**Patent**

and return the application variables to the graphical user interface, via the second module.

8. (Previously Presented) The system of Claim 6, wherein the graphical user interface is further configured to receive an input identifying requested application values to be displayed.

9. (Currently Amended) The system of Claim 1, wherein the first COBOL program module is further configured as a socket server through the technical layer and wherein the second module is further configured as a socket client such that the first COBOL program module and second module[[s]] communicate via a socket connection.

10. (Currently Amended) The system of Claim 1, wherein the first COBOL program module reads application values stored in the shared memory area by the at least one application while the at least one application is running.

11. (Currently Amended) The system of Claim 10, wherein the first COBOL program module reads application values stored in the memory area by the at least one application without interfering with the operation of the at least one application.

**Atty Docket: IDF 2584 (4000-16100)**

**Patent**

12. (Currently Amended) A method of non-intrusively monitoring operation of an application, comprising:

~~running an application in a real time manner;~~

~~creating a memory area;~~

~~generating, by the application, application values during operation of the application;~~

~~writing, by the application, the application values in the memory area during the operation of the application;~~

attaching, by a COBOL (Common Business Oriented Language) program monitor through a technical layer, to a memory area created by the application, wherein the memory area includes application values generated by the application and written by the application to the memory area during operation of the application in a real-time manner, wherein the COBOL program monitor is programmed with COBOL that does not provide support for features to attach to the memory area, and wherein the technical layer comprises a memory routine that includes a function for attaching to a memory area that enables the COBOL program monitor to attach to the memory area;

reading, by ~~[[a]]~~the COBOL program monitor, the memory area used by the application to obtain the application values, wherein at least one of the application values is not output by the application; and

displaying the application values read from the memory area.

*Atty Docket: IDF 2584 (4000-16100)**Patent*

13. (Currently Amended) The method of Claim 12, further comprising:

requesting, by a client, the application values from the COBOL program monitor;  
and

communicating the application values variables from the COBOL program monitor  
to the client.

14. (Currently Amended) The method of Claim 13, further comprising:

requesting application values;

~~running a plurality of applications in a real-time manner;~~

~~generating application values stored in one or more memory areas during  
operation of the plurality of applications;~~

attaching, by the COBOL program monitor through the technical layer, to a plurality  
of memory areas that include application values generated by a plurality of  
applications during operation of the plurality of applications in a real-time  
manner.

reading, by the COBOL program monitor, from the plurality of the one or more  
memory areas used by the plurality of applications to obtain the application  
values generated by the plurality of applications; and

displaying the requested application values generated by the plurality of  
applications.



*Atty Docket: IDF 2584 (4000-16100)**Patent*

15. (Currently Amended) The method of Claim 14, wherein the memory area is further defined as a block of memory ~~and wherein the monitor reads at least some of the application variables stored in the block of memory.~~

16. (Currently Amended) The method of Claim 13, further comprising providing a memory manager and wherein the COBOL program monitor registers with the memory manager to obtain a location of the memory area used by the application to store the application values.

17. (Currently Amended) The method of Claim 13, further comprising:

~~generating new application values by the application stored in the memory area, at least one of the new application values defined as a new value for a variable of the application;~~

requesting, by the client, that the monitor re-read the application values stored in the memory area, at least some of the application values including new application values generated by the application and stored in the memory area, at least one of the new application values defined as a new value for a variable of the application;

re-reading, by the COBOL program monitor, the memory area to obtain the new application values.

18. (Currently Amended) The method of Claim 17, wherein the COBOL program monitor reads the application values while the application is running.

***Atty Docket: IDF 2584 (4000-16100)******Patent***

19. (Currently Amended) The method of Claim 13, wherein the COBOL program monitor is configured as a socket server through the technical layer and wherein the client is configured as a socket client such that the communication between the COBOL program monitor and client is via a socket connection.

20. (Original) The method of Claim 12, wherein the application values are further defined as a variable of the application and a value of the variable.

*Atty Docket: IDF 2584 (4000-16100)**Patent*

21. (Currently Amended) A system for non-intrusively monitoring variables during operation of an application, comprising:

a compile listing stored on a non-transitory computer-readable medium having an address map with an offset associated with each of a plurality of variables of an application; and

a COBOL (Common Business Oriented Language) program module stored on a non-transitory computer-readable medium that is programmed to perform[[s]] reading of the compile listing and obtaining the offset of at least one of the plurality of variables of the application, the COBOL program module further programmed to perform[[s]] attaching through a technical layer to an address space used by the application during real-time operation of the application to obtain a value for one or more of the plurality of variables written to the address space by the application during the real-time operation of the application using the offset, wherein the COBOL program module is programmed with COBOL that does not provide support for features to attach to the address space, and wherein the technical layer comprises a memory routine that includes a function for attaching to an address space that enables the COBOL program module to attach to the address space.

22. (Currently Amended) The system of Claim 21, wherein the COBOL program module is further configured-programmed to read the compile listing and convert at least one of the plurality of variables to the associated offset.

***Att'y Docket: IDF 2584 (4000-16100)***

***Patent***

23. (Currently Amended) The system of Claim 21, wherein the COBOL program module is further configured to search the compile listing and display the plurality of variables of the application for selection by a user.

24. (Currently Amended) The system of Claim 23, wherein the COBOL program module is responsive to selection by the user of one of the plurality of variables to obtain the value for the selected one of the plurality of variables using the offset to locate the value of the variable in the address space.

25. (Currently Amended) The system of Claim 24, wherein the COBOL program module is further configured to display the selected one of the plurality of variables.

26. (Currently Amended) The system of Claim 21, wherein the address space is further defined as a memory space and wherein the COBOL program module attaches, using a socket layer of the technical layer, to the memory space used by the application.

27. (Currently Amended) The system of Claim 26, wherein the COBOL program module attaches, using the offset, to the memory space used by the application via an operating system service accessible through the technical layer.

*Atty Docket: IDF 2584 (4000-16100)**Patent*

28. (Currently Amended) The system of Claim 21, wherein the COBOL program monitor is further configured, using the compile listing, to query the address map for one or more of the plurality of variables of the application.

29. (Canceled)

30. (Currently Amended) The system of Claim 21, wherein the COBOL program module is further configured to attach to the memory space where the application is operating and overwrite the value for one or more of the plurality of variables using the offset.

31. (Currently Amended) The system of Claim 21, wherein the COBOL program module comprises:

a reader component ~~configured~~ programmed to perform reading ~~read~~ the compile listing and further ~~configured to perform converting~~ programmed to convert at least one of the plurality of variables of the application to the associated offset; and

a search component ~~[[that]]~~ programmed to receive ~~performs~~ ~~receiving~~ the associated offset of the at least one of the plurality of variables from the reader component, the search component ~~configured to perform attaching~~ programmed to attach to the application and ~~further operable to locate~~ the value of the at least one of the plurality of variables using the offset.

*Atty Docket: IDF 2584 (4000-16100)**Patent*

32. (Currently Amended) The system of Claim 21, further comprising a display component operably coupled to the COBOL program module to perform receiving the value for the one or more of the plurality of variables, the display component configured to perform displaying the value.

33. (Previously Presented) The system of Claim 32, wherein the display component is configured to employ the value to display a heartbeat.

34. (Previously Presented) The system of Claim 32, wherein the display component is configured to employ the value to display as a percentage complete.

*Atty Docket: IDF 2584 (4000-16100)**Patent*

35. (Currently Amended) A system for non-intrusively monitoring COBOL application values, the system comprising:

a COBOL (Common Business Oriented Language) program stored on a non-transitory computer-readable medium that is programmed to create[[s]] a shared memory area through a technical layer, generate[[s]] program values, and store[[s]] the program values in the shared memory area during real-time operation of the COBOL program, wherein the COBOL program is programmed with COBOL that does not provide support for features to create the shared memory area, and wherein the technical layer comprises a shared memory routine that includes a function for creating shared memory that enables the COBOL program module to create the shared memory area; and

a COBOL program monitor module stored on a non-transitory computer-readable medium that is programmed to share[[s]] the shared memory area with the COBOL program through the technical layer, and the COBOL program monitor module is programmed to read[[s]] the program values stored in the shared memory area by the COBOL program during real-time operation of the COBOL program, wherein the COBOL program monitor is programmed with COBOL that does not provide support for features to share the shared memory area, and wherein the technical layer comprises a shared memory routine that includes a function for sharing shared memory that enables the COBOL program module to share the shared memory area.

36. (Currently Amended) The system of Claim 35, further comprising:

a second COBOL program configured to generate second program values and store the program values in the shared memory area during real-time operation of the

*Atty Docket: IDF 2584 (4000-16100)**Patent*

second COBOL program, and wherein the COBOL program monitor module is further configured to read the second program values stored in the shared memory area by the second COBOL program.

37. (Currently Amended) The system of Claim 35, further comprising:

a second memory area; and

a second COBOL program configured to generate second program values and store the program values in the second memory area during real-time operation of the second COBOL program, and wherein the COBOL program monitor module is further configured to read the second program values stored in the second memory area by the second COBOL program.

38. (Previously Presented) The system of Claim 35, further comprising:

a user interface configured to monitor and display the application values; and

a client application in communication with the user interface and the COBOL program monitor module, the client application configured to request the program variables of the COBOL program from the COBOL program monitor module and provide the program variables to the user interface for display via the user interface responsive to a request from the user interface.

39. (New) The system of claim 1, wherein the second module is a second COBOL program module, wherein the second COBOL program module and the first COBOL



***Atty Docket: IDF 2584 (4000-16100)******Patent***

program module are in communication using a socket connection established through the technical layer, wherein the first COBOL program module and the second COBOL program module are programmed with COBOL that does not provide support for socket connections, and wherein the technical layer comprises a socket routine that enables the first COBOL program module and the second COBOL program module to manage socket communications through operating system calls.

*Any Docket: IDF 2584 (4000-16100)**Patent***CONCLUSION**

Applicant respectfully submits that the present application is in condition for allowance for the reasons stated above. If the Examiner has any questions or comments or otherwise feels it would be helpful in expediting the application, the Examiner is encouraged to telephone the undersigned at (972) 731-2286.

The Commissioner is hereby authorized to charge payment of any further fees associated with any of the foregoing papers submitted herewith, or to credit any overpayment thereof, to Deposit Account No. 21-0765, Sprint.

Respectfully submitted,

Date: November 9, 2010

/Brian C. Genco/  
Brian C. Genco  
Reg. No. 58,096  
Michael W. Piper  
Reg. No. 39,800

Conley Rose, P.C.  
5601 Granite Parkway, Suite 750  
Plano, Texas 75024  
(972) 731-2288  
(972) 731-2289 (facsimile)

ATTORNEY FOR APPLICANT